AMENDMENTS TO THE CLAIMS

A1

1. (currently amended) A computer program product encoding a computer program for executing on a computer system a computer process for identifying a root set of pointers to a heap in a call stack using ~~building~~ compact garbage collection tables adapted for use in reclaiming memory from [[a]]the heap during runtime, the computer process comprising:

generating a first call site table storing call site identifiers, each call site identifier identifying the location of a call site in a computer program;

generating a final descriptor table storing a set of unique descriptors to be associated with one or more stack frames by a garbage collector ~~at least one~~, each unique descriptor describing a set of registers containing pointers to the heap and a set of offsets relative to a location within each of the one or more stack frame containing pointers to the heap; ~~location of a pointer into the heap; and~~

generating a descriptor reference table associated with the first call site table, each entry in the descriptor reference table mapping a call site identifier in the first call site table to one of the unique descriptors in the final descriptor table;

traversing the call stack and associating a first stack frame in the call stack with a first call site identifier using the first call site table;

associating the first call site identifier with a first unique descriptor in the final descriptor table using the descriptor reference table; and

using the first unique descriptor to identify the set of registers containing pointers to the heap associated with the first stack frame and the locations of pointers within the first stack frame containing pointers to the heap.

2. (original) The computer program product of claim 1 wherein at least two call site identifiers are mapped to the same unique descriptor in the final descriptor table.

3. (cancelled)

25

A1

3 ~~4~~. (original) The computer program product of claim 1 wherein the operation of generating a first call site table comprises:

storing return addresses for one or more call sites into the call site table.

4 ~~5~~. (original) The computer program product of claim 1 wherein the operation of generating a final descriptor table comprises:

generating an initial descriptor table including at least two identical descriptors, each descriptor in the initial descriptor table corresponding with a call site identifier in the call site table;

copying each descriptor from the initial descriptor table to the final descriptor table, if the descriptor is not identical to another descriptor already copied to the final descriptor table.

5 ~~6~~. (original) The computer program product of claim 1 wherein the operation of generating a descriptor reference table comprises:

generating a table pair including a second call site table and an initial descriptor table, the initial descriptor table storing descriptors that include at least two identical descriptors;

sorting the table pair based on the descriptors in the initial descriptor table to provide a sorted table pair;

traversing sequentially through the descriptors in the sorted table pair to associate a reference to each call site in the second call site table, the reference being modified when a unique descriptor is encountered in the initial descriptor table;

identifying each call site identifier in the second call site table to which each reference is associated; and

storing each reference into the descriptor reference table in association with the call site identifier identified in the identifying operation.

6 ~~7~~. (original) The computer program product of claim 5 ~~6~~ wherein the reference is an ordinal identifier of unique descriptors being processed during the traversing operation.

26

A

A1

~~8~~ 7. (original) The computer program product of claim ~~7~~ 6 wherein the traversing operation comprises:

incrementing the ordinal identifier when a unique descriptor is encountered in the initial descriptor table.

~~9~~ 8. (original) The computer program product of claim ~~6~~ 5 wherein the reference includes a pointer to one of the unique descriptors stored in the final descriptor table.

~~10~~ 9. (original) The computer program product of claim ~~9~~ 8 wherein the traversing operation comprises:

designating as the reference a new pointer to one of the unique descriptors in the final descriptor table when a unique descriptor is encountered in the initial descriptor table.

27

K

A1

11. (currently amended) A computer-readable medium having stored thereon compact garbage collection tables for identifying elements of a root set used in reclaiming memory from a heap during runtime, the compact garbage collection tables comprising:

a call site table storing call site identifiers;

a final descriptor table storing a set of unique descriptors, each unique descriptor describing a set of registers to be associated with at least one stack frame of a call stack, the set of registers containing pointers to the heap, and a set of offsets relative to a location within the at least one stack frame containing pointers to the heap; and

a descriptor reference table associated with the call site table, each entry in the descriptor reference table mapping a call site identifier in the call site table to a unique descriptor in the descriptor table.

12. (original) The computer readable medium of claim 11 wherein at least two call site identifiers of the call site table are mapped to the same unique descriptor in the descriptor table.

13. (cancelled).

A1

10 ~~14~~. (currently amended) A method of building compact garbage collection tables adapted for use in reclaiming memory from a heap during runtime, the method comprising:

generating a first call site table storing call site identifiers including a plurality of first call site identifiers, each first call site identifier identifying a call site in a different stack frame, each first call site having a first set of pointers into the heap, the first set of pointers located in an identical first set of registers associated with the different stack frame and an identical first set of offsets relative to a first location within the different stack frame;

generating a final descriptor table storing a set of unique descriptors, ~~at least one unique~~ including only one first descriptor describing the first set of registers to be associated with an unidentified stack frame and the first set of offsets relative to the first location within the unidentified stack frame; ~~a location of a pointer into the heap,~~; and

generating a descriptor reference table associated with the first call site table, each entry in the descriptor reference table mapping a call site identifier in the first call site table to one of the unique descriptors in the final descriptor table, wherein the descriptor reference table includes an entry for each first call site identifier mapping the first call site identifier to the first descriptor in the final descriptor table.

11 ~~15~~. (currently amended) The method of claim 10 ~~14~~ wherein ~~at least two call site identifiers of the first call site table are mapped to the same unique descriptor in the final descriptor table~~:

the first call site table includes second call site identifiers, each second call site identifier and first call site identifier identifying a call site in a different stack frame, each second call site having a second set of pointers into the heap, the second set of pointers located in an identical second set of registers associated with the different stack frame and an identical second set of offsets relative to a first location within the different stack frame;

the set of unique descriptors includes only one second descriptor describing the second set of registers to be associated with an unidentified stack frame and the second set of offsets relative to the first location within the unidentified stack frame; and

wherein the descriptor reference table includes an entry for each second call site identifier mapping the second call site identifier to the second descriptor in the final descriptor table.

16. (cancelled).

28 A

A\

12 ~~17.~~ (original) The method of claim 10 ~~14~~ wherein the operation of generating a first call site table comprises:

storing return addresses for one or more call sites into the call site table.

13 ~~18.~~ (currently amended) The method of claim 10 ~~14~~ wherein the operation of generating a final descriptor table comprises:

generating an initial descriptor table including ~~at least two identical~~ a plurality of identical first descriptors, each first descriptor in the initial descriptor table corresponding with a different first call site identifier in the call site table; and

copying each descriptor, including the first descriptor, from the initial descriptor table to the final descriptor table, if the descriptor is not identical to another descriptor already copied to the final descriptor table.

14 ~~19.~~ (currently amended) The method of claim 10 ~~14~~ wherein the operation of generating a descriptor reference table comprises:

generating a table pair including a second call site table and an initial descriptor table, the initial descriptor table storing descriptors that include ~~at least two identical~~ the plurality of identical first descriptors;

sorting the table pair based on the descriptors in the initial descriptor table to provide a sorted table pair;

traversing sequentially through the descriptors in the sorted table pair to associate a reference to each call site in the second call site table, the reference being modified when a unique descriptor is encountered in the initial descriptor table;

identifying each call site identifier in the second call site table to which each reference is associated; and

storing each reference into the descriptor reference table in association with the call site identifier identified in the identifying operation.

15 ~~20.~~ (original) The method of claim 14 ~~19~~ wherein the reference is an ordinal identifier of the descriptor and call site identifier being processed during the traversal of the sorted table pair.

29

A

A1

16 ~~21~~. (original) The method of claim 15 ~~20~~ wherein the traversing operation comprises:

incrementing the ordinal identifier when a unique descriptor is encountered in the initial descriptor table.

17 ~~22~~. (original) The method of claim 14 ~~19~~ wherein the reference includes a pointer to one of the unique descriptors stored in the final descriptor table.

18 ~~23~~. (original) The method of claim 17 ~~22~~ wherein the traversing operation comprises:

designating as the reference a new pointer to one of the unique descriptors in the final descriptor table when a unique descriptor is encountered in the initial descriptor table.

30

A

A1

24. (currently amended) A computer program product encoding a computer program for executing on a computer system a computer process for identifying elements of a root set for garbage collection using compact garbage collection tables, the computer process comprising:

locating a call site identifier in a call site table;

identifying a descriptor reference in a descriptor reference table, the descriptor reference being associated with the call site identifier;

identifying a descriptor referenced by the descriptor reference, the descriptor reference mapping between the call site identifier and the descriptor, the descriptor being one descriptor of a set of unique descriptors in a descriptor table and describing a set of registers associated with an unidentified stack frame of a call stack and a set of offsets relative to a location within the unidentified stack frame containing pointers to the heap; and

accessing the descriptor to determine the elements of the root set for a call site identified by the call site identifier for garbage collection.

25. (original) The computer program product of claim 24 wherein the descriptor is mapped to at least two call site identifiers in the call site table.

26. (cancelled)

27. (original) The computer program product of claim 24 wherein the descriptor reference includes an index into the descriptor table.

28. (original) The computer program product of claim 24 wherein the descriptor reference includes a pointer into the descriptor table.

A

A1

29. (currently amended) A runtime system for identifying elements of a root set for a garbage collection using compact garbage collection tables, the runtime system comprising:

a garbage collector traversing a call stack consisting of stack frames, accessing a call site table storing call site identifiers, and associating a call site identifier with each stack frame;

a descriptor table storing a set of unique descriptors identifying elements of a root set of pointers into a heap, each unique descriptor describing a location of a pointer into the heap relative to an unidentified stack frame, and

a descriptor reference table associated with the call site table to identify elements of a root set, each entry in the descriptor reference table mapping one of the call site identifiers in the call site table to a descriptor in the descriptor table, at least one descriptor in the descriptor table being mapped to a plurality of call site identifiers.

30. (cancelled).

A

A1

31. (currently amended) A method of identifying elements of a root set for garbage collection using compact garbage collection tables, the method comprising:

locating a call site identifier in a call site table;

identifying a descriptor reference in a descriptor reference table, the descriptor reference being associated with the call site identifier;

identifying a descriptor referenced by the descriptor reference, the descriptor reference mapping between the call site identifier and the descriptor, the descriptor being one descriptor of a set of unique descriptors in a descriptor table and describing a set of registers to be associated with a stack frame containing pointers to the heap and a set of offsets relative to a location within the stack frame; and

accessing the descriptor to determine the elements of the root set for a call site identified by the call site identifier for garbage collection.

32. (original) The method of claim 31 wherein the descriptor is mapped to at least two call site identifiers in the call site table.

33. (cancelled).

A